Examples of Decidable languages:

- $\text{A}_{\text{DFA}} = \{ \langle B, w \rangle : B \text{ is a DFA that accepts input string } w \} \text{ is decidable.}

  TM M that decide $\text{A}_{\text{DFA}}$

  $M = " \text{On input } \langle B, w \rangle, \text{ where } B \text{ is a DFA and } w \text{ is a string:}"

  1. Simulate $B$ on input $w$.
  2. If the simulation ends in an accepting state, accept. If it ends
     in a non-accepting state, reject."

- $\text{A}_{\text{NFA}} = \{ \langle B, w \rangle : B \text{ is an NFA that accepts string } w \} \text{ is decidable.}

  $N_1 = " \text{On input } 2B, w:\n
  1. Simulate NFA $B$ on input $w$.
  2. If simulation ends on an accepting state, accept. If it ends in
     a non-accepting state, reject."

  $N_1$ is a non-deterministic Turing machine.

  $N_2 = " \text{On input } 2B, w:\n
  1. Convert NFA $B$ to an equivalent DFA $C$.
  2. Run TM $M$ (from previous example) on input $\langle C, w \rangle$.
  3. Accept if $M$ accepts, reject if $M$ rejects."

  $N_2$ is deterministic TM.

- $\text{A}_{\text{REG}} = \{ \langle R, w \rangle : R \text{ is a regular expression and } R \text{ generates } w \} \text{ is decidable.}

  $D = " \text{On input } \langle R, w \rangle:\n
  1. Convert $R$ into NFA $A$.
  2. Run TM $N_2$ on $\langle A, w \rangle$.
  3. If $N_2$ accepts, accept; if $N_2$ rejects, reject."
E_DFA = \{ <A> : A is a DFA and L(A) = \emptyset \} is decidable.

T = " On input <A>:
1. Mark the start state of A.
2. Repeat until no new states are marked:
   a) Mark any state that has a transition coming into it from any state that is already marked.
3. If no accept state is marked, accept; otherwise reject."

E_DFA = \{ <A,B> : A and B are DFA and L(A) \cap L(B) \}

Blue: Symmetric Difference of L(A) and L(B).
L_1 = (L(A) \cap \overline{L(B)}) \cup (L(B) \cap \overline{L(A)})

We know from properties of regular languages that L_1 is regular. Consider a DFA C for L_1.

F = " On input <A,B>:
1. Construct DFA C as described.
2. Run TM T from previous example on input <C>.
3. If T accepts, accept; if T rejects, reject."

A_CE = \{ <G,w> : G is a CFG that generates string w \} is decidable.

S = " On input <G,w>:
1. Convert G to an equivalent grammar in Chomsky normal form.
2. Let all derivations with 2n-1 steps, where n=|w|. If no,,
   let all derivations of length \leq n.
3. If any of these derivations accept w, accept; if not reject."

Fact: If grammar G in Chomsky normal form, for any string
w (n=|w|), any derivation of w has 2n-1 steps.

E_CE = \{ <G> : G is a CFG and L(G) = \emptyset \} is decidable.
R = " On input <6,7>:
1. Mark all terminal symbols in 6.
2. Repeat until no new variable get marked:
   a) Mark any variable A where 6a has a rule
      A → U₁U₂ ... Uₙ and
      each symbol Uᵢ,...,Uₙ has already been marked.
3. If the start variable is not marked, accept; otherwise reject.

Idea: Determine for each variable, in particular also for the start variable, if it is capable of generating a string of terminals.

* Every context-free language A is decidable.

Let 6 be a context-free grammar for A.

M₆ = " On input w:
   1. Run TM S (as defined previously) on input <6,w>.
   2. If S accepts, accept; if S rejects, reject."

EQ₆₆ = \{<6,7> : Mₑ and M₆ on L₆ₑ and L₆₆ = L₆\}.

Cannot use arguments as for Regular languages since context-free languages are not closed under complementation and intersection.

In fact, EQ₆₆ is NOT decidable.

Undecidable languages:

Aₑₘ = \{<M,w> : M is a TM and M accepts w\}.

Aₑₘ is recognizable.

U = " On input <M,w>:
   1) Run M on input w.
      " A * M ... = P * M * " "

TOC Page 3
2) Accept if $M$ accepts. Reject if $M$ rejects.

$U$ is called the 'Universal Turing Machine'.

We will show $A_m$ is not decidable.

**The Diagonalization Method:**

A function $f : A \rightarrow B$ is one-to-one if

$$\forall a_1, a_2 \in A, a_1 \neq a_2 \Rightarrow f(a_1) \neq f(a_2)$$

A function $f : A \rightarrow B$ is onto if

$$\forall b \in B, \exists a \in A \text{ s.t. } f(a) = b$$

Function $f$ is called a 'correspondence' if it is one-to-one and onto.

**George Cantor (1873):** Two sets $A$ and $B$ are of same size iff there exists correspondence $f : A \rightarrow B$.

Examples:

- $\{1, 2\}$ some size $\{5, 8\}$.
- Set of natural numbers $N = \{1, 2, 3, \ldots\}$ some size as set of even numbers $E = \{2, 4, 6, 8, \ldots\}$.
- Set of rational numbers $\mathbb{Q} = \{\frac{m}{n} : m, n \in \mathbb{Z}, n \neq 0\}$ some size as $N$.

**Def:** (Countable set): A set is countable if it is finite or has the same size as $N$.

A set which is not countable is called uncountable.

Thus, the set of real numbers $\mathbb{R}$ is uncountable.

**Proof:** Assume for contradiction that $\mathbb{R}$ is countable. Let $f : N \rightarrow \mathbb{R}$ be a correspondence between $N$ and $\mathbb{R}$.

Let $x = 0.x_1x_2x_3 \ldots$
\[ x \notin f[1, 0.9^2] \text{. Then } x \notin \text{Range of } f. \]

Hence \( f \) is not one-to-one and hence contradiction.

Thus: Some (almost all) languages are not recognizable.

Proof: The set of all Turing machines is countable.

The set of all languages is uncountable.

Thus: \( \text{Arm} = \{ \langle M, w \rangle : M \text{ in a TM and } M \text{ accept } w \} \) is undecidable.

Proof: Assume for contradiction that \( \text{Arm} \) is decidable. Let \( H \) be a decider for \( \text{Arm} \):

\[ H(\langle M, w \rangle) = \begin{cases} 
\text{accept if } M \text{ accept } w \\
\text{reject otherwise.}
\end{cases} \]

Let \( D \): "On input \( \langle M \rangle \):

1. Run \( H \) on \( \langle M, \langle \rangle \rangle \).
2. Accept if \( H \) reject; reject if \( H \) accept."

\[ D(\langle M \rangle) = \begin{cases} 
\text{reject if } M \text{ accept } \langle \rangle \\
\text{accept if } M \text{ not accept } \langle \rangle.
\end{cases} \]

\[ D(\langle \emptyset \rangle) = \begin{cases} 
\text{reject if } D \text{ accepts } \langle \emptyset \rangle \\
\text{accept if } D \text{ not accept } \langle \emptyset \rangle.
\end{cases} \]

Contradiction! \( \square \)

Definition (co-Turing-recognizable): Language \( A \) is co-Turing-recognizable if \( \overline{A} \) is Turing-recognizable.

Thm: A language \( B \) is decidable if \( H \) is Turing-recognizable and co-Turing-recognizable.

Proof: \((\Rightarrow)\) Easy to argue.

\((\Leftarrow)\) Let \( M_1 \) be recognizer of \( B \) and \( M_2 \) be recognizer of \( \overline{B} \).

\( M_1 : " \) On input \( w \):

1. Run both \( M_1 \) and \( M_2 \) on input \( w \) in parallel.
2. If \( M_1 \) accept, accept; if \( M_2 \) accept, reject.\( \square \)
Geoffrey: \( \overline{Am} \) is \textbf{Not} \textsc{turing-recognizable}.

Paul: We know \( \overline{Am} \) is \textsc{turing-recognizable}, and not decidable. \( \square \).